Docket No. 1076.1060 (JDH)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:	)	
	)	
Satoshi SAKAMOTO, et al.	)	
	)	Group Art Unit: Unassigned
Serial No.: To be assigned	)	
	)	Examiner: Unassigned
Filed: December 28, 2000	)	

tc910 U.S. PTO
09/748862
12/28/00

For: **METHOD FOR CONTROLLING MULTITHREADING**

SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN APPLICATION IN ACCORDANCE WITH THE REQUIREMENTS OF 37 C.F.R. §1.55

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

In accordance with the provisions of 37 C.F.R. §1.55, the applicants submit herewith a certified copy of the following foreign application:

Japanese Patent Application No. 2000-016070
Filed: January 25, 2000.

It is respectfully requested that the applicants be given the benefit of the foreign filing date as evidenced by the certified papers attached hereto, in accordance with the requirements of 35 U.S.C. §119.

Respectfully submitted,

STAAS & HALSEY LLP

By: _____
James D. Halsey, Jr.
Registration No. 22,729

700 11th Street, N.W., Ste. 500
Washington, D.C. 20001
(202) 434-1500
Date: December 28, 2000

【書類名】　特許願

【整理番号】　9951413

【提出日】　平成12年 1月25日

【あて先】　特許庁長官殿

【国際特許分類】　G06F　9/46

【発明の名称】　マルチスレッド制御方法、マルチスレッド制御装置、及び記録媒体

【請求項の数】　6

【発明者】

【住所又は居所】　神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内

【氏名】　坂本　智

【発明者】

【住所又は居所】　神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内

【氏名】　岩崎　正則

【発明者】

【住所又は居所】　神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内

【氏名】　坪井　晃

【特許出願人】

【識別番号】　000005223

【氏名又は名称】　富士通株式会社

【代理人】

【識別番号】　100068755

【住所又は居所】　岐阜市大宮町２丁目１２番地の１

【弁理士】

【氏名又は名称】　恩田　博宣

【電話番号】　058-265-1810

【選任した代理人】

【識別番号】　100105957

【住所又は居所】　東京都渋谷区代々木二丁目１０番４号　新宿辻ビル８階

【弁理士】

【氏名又は名称】　恩田　誠

【電話番号】　03-5365-3057

【手数料の表示】

【予納台帳番号】　002956

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【包括委任状番号】　9909792

【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　マルチスレッド制御方法、マルチスレッド制御装置、及び記録媒体

【特許請求の範囲】

【請求項１】　複数のスレッドによって並列処理を行うマルチスレッド制御方法において、

並列処理を行っている実行中スレッドの数と待機状態にある待機スレッドの数を監視し、両スレッド数に基づいて不要な数の待機スレッドを回収する

ことを特徴とするマルチスレッド制御方法。

【請求項２】　所定時間毎に、前記待機スレッドの数と、前記所定時間内の実行中スレッドの数の最大値、平均値、及びそれらに所定の係数を乗じた値のうちの何れか一つからなる必要数とを比較し、前記待機スレッドの数が前記必要数より多い場合に該必要数まで前記待機スレッドを回収することを特徴とする請求項１記載のマルチスレッド制御方法。

【請求項３】　複数のスレッドによって並列処理を行うマルチスレッド制御装置において、

前記複数のスレッドのスレッド情報が記憶されるスレッド情報管理部と、

前記スレッド情報管理部に記憶された待機状態にあるスレッドの数に基づいてスレッドの生成要求、待機状態にあるスレッドに対して処理の実行要求を行うスレッド実行処理部と、

前記スレッド情報管理部に記録された並列処理を行っている実行中スレッドの数と待機状態にある待機スレッドの数に基づいて不要な数の待機スレッドを回収するスレッド回収処理部と

を備えたことを特徴とするスレッド制御装置。

【請求項４】　前記スレッド回収処理部は、

所定時間毎に、前記待機スレッドの数と、前記所定時間内の実行中スレッドの数の最大値、平均値、及びそれらに所定の係数を乗じた値のうちの何れか一つからなる必要数とを比較する手段と、

前記待機スレッドの数が前記必要数より多い場合に該必要数まで前記待機スレ

ッドを回収する手段と

を備えたことを特徴とする請求項３記載のスレッド制御装置。

【請求項５】　複数のスレッドによって並列処理を行うマルチスレッド制御プログラムが記録されたコンピュータ読み取り可能な記録媒体であって、

前記プログラムは、

並列処理を行っている実行中スレッドの数と待機状態にある待機スレッドの数を監視するステップと、

両スレッド数に基づいて不要な数の待機スレッドを回収するステップ

を実行する、記録媒体。

【請求項６】　前記プログラムは、

所定時間毎に、前記待機スレッドの数と、前記所定時間内の実行中スレッドの数の最大値、平均値、及びそれらに所定の係数を乗じた値のうちの何れか一つからなる必要数とを比較するステップと、

前記待機スレッドの数が前記必要数より多い場合に該必要数まで前記待機スレッドを回収するステップ

を実行する、請求項５記載の記録媒体。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、複数のスレッドによって並列処理を行うシステムにおいて、プログラムの実行速度の向上とシステム資源の有効活用を図ることができるマルチスレッド制御方法、その装置、及び記録媒体に関するものである。

【０００２】

近年、科学技術計算等のプログラムを高速に実行するシステムのオペレーティングシステムプログラムには並列動作機構が備えられているものがある。このようなオペレーティングシステムは、プログラムを所定の処理で分割し、各処理をシステム資源を割り当てたスレッドにて平行実行することで、プログラムの実行速度を高くしている。このようなシステムにおいて、プログラム実行速度の高速化とシステム資源の有効活用が求められている。

【０００３】

【従来の技術】

従来、科学技術計算等のプログラムを実行するシステムでは、プログラムの並列実行などによってプログラムの実行速度の高速化を図る等を目的にスレッドが用いられる。システムのオペレーティングシステム（ＯＳ）プログラムは、アプリケーションに応じてスレッドを生成する。生成した各スレッドにシステム資源を割り当てることで、各スレッドに対応させたプログラムを並列実行し、プログラムの実行速度を高くしている。

【０００４】

しかし、ＯＳプログラムのスレッド生成には時間がかかるため、一度生成したスレッドを削除することなく再利用することで、システムにおける処理の高速化を図ることが行われている。

【０００５】

詳述すると、プログラムの実行を終了したスレッドを終了させずに待機スレッドとしてテーブルに登録する。そして、スレッドの生成要求があった場合には、テーブルに登録した待機スレッドを用いてプログラムを実行する。これにより、スレッド生成時間を短縮・削減する。

【０００６】

【発明が解決しようとする課題】

従来のマルチスレッド制御方法では、生成されたスレッドの数だけ待機スレッドとして管理されており、実際に動作しているスレッドが少ない状態でも、不要な待機スレッドがシステムに沢山存在している。

【０００７】

待機スレッドにもシステム資源が割り当てられている。即ち、待機スレッドは、その待機中でもメモリを使用している。このように、待機スレッドが無駄にメモリ資源を使用している状況に陥ることがある。

【０００８】

例えば、ある時刻において２００個のスレッドが並列に動作し、その後２個のスレッドしか動作しない場合、１９８個の待機スレッドがメモリを使用すること

になる。システム資源の少ないシステムにおいて、このような状況はシステムの運営上好ましくなく、他のプログラムの実行速度にも大きな影響を与える。

【０００９】

本発明は上記問題点を解決するためになされたものであって、その目的はプログラム実行速度の高速化とシステム資源の有効活用を図ることのできるマルチスレッド制御方法その装置、及び記録媒体を提供することにある。

【００１０】

【課題を解決するための手段】

上記目的を達成するため、本発明は、並列処理を行っている実行中スレッドの数と待機状態にある待機スレッドの数を監視し、両スレッド数に基づいて不要な数の待機スレッドを回収する。これにより、無駄にメモリ資源を使用している待機スレッドを終了させ、システム資源の有効活用が図られる。

【００１１】

また、所定時間毎に、前記待機スレッドの数と、前記所定時間内の実行中スレッドの数の最大値、平均値、及びそれらに所定の係数を乗じた値のうちの何れか一つからなる必要数とを比較し、前記待機スレッドの数が前記必要数より多い場合に該必要数まで前記待機スレッドを回収する。これにより、稼働状況に応じた必要数にまで待機スレッドの数を低減してシステム資源の有効活用が図られる。

【００１２】

【発明の実施の形態】

以下、本発明を具体化した一実施の形態を図１～図６に従って説明する。

図１は、一実施形態のマルチスレッド制御装置の概略構成図である。

【００１３】

マルチスレッド制御装置１は、アプリケーションプログラムを実行するシステム（計算機）に備えられ、オペレーティングシステム（ＯＳ）２にスレッド生成を要求し、ＯＳ２から受け取ったスレッドを用いてプログラムの並列可能な部分プログラム（例えばＤＯループやサブルーチン）を並列実行する。

【００１４】

マルチスレッド制御装置１は、部分プログラムの終了後、不要となったスレッ

ドを待機スレッドとして登録する。次に、プログラムの実行要求があると、マルチスレッド制御装置１は、待機スレッドの状態を見て、待機スレッドがあればそれを用いて部分プログラムを実行し、待機スレッドが無い場合にはＯＳ２にスレッド生成を要求する。

【００１５】

マルチスレッド制御装置１は、プログラム実行中のスレッドの数と待機スレッドの数を監視し、実行中スレッド数に基づいて、必要数以上の待機スレッドを回収する。例えば、マルチスレッド制御装置１は、最大実行中スレッド数と待機スレッド数を用いて、待機スレッドの回収処理を実行する。

【００１６】

詳述すると、マルチスレッド制御装置１は、実行中のスレッドの数を所定期間監視し、その期間内の最大値（最大実行中スレッド数）をその時に於ける待機スレッドの必要数とし、それと現在管理している待機スレッドの数とを比較する。そして、マルチスレッド制御装置１は、最大実行中スレッド数よりも待機スレッド数が大きい場合、最大実行中スレッド数まで待機スレッドを回収する。即ち、マルチスレッド制御装置１は、待機スレッド数が最大実行中スレッド数と一致するまで待機スレッドを終了させる。

【００１７】

このように、無駄にシステム資源を使用している待機スレッドを回収することで、実行中のスレッドに与える影響を少なくする。

また、最大実行中スレッド数は、実行しているプログラムの状態、即ち、システムの稼働状況に応じて変化する。従って、最大実行中スレッド数まで待機スレッドを回収することで、システムの稼働状況に応じて待機スレッド数を動的に変化させている。

【００１８】

更に、所定期間実行中スレッドを監視した最大実行中スレッド数にまで待機スレッドを回収することで、不意なスレッドの増加に対応することができる。即ち、実行中にスレッド数が一時的に低下しても、最大実行中スレッド数だけ待機スレッドを待機させているため、待機スレッドが不足することが無く、プログラム

を直ぐに実行することができる。

【００１９】

一方、待機スレッド数が実行中スレッド数と等しくなるように待機スレッドを回収する方法では、実行中スレッド数の一時的な低下に対応して待機スレッド数を低減させるため、次にプログラムに応じてスレッドが必要となった場合に待機スレッドが不足してスレッドの生成要求をＯＳ２に行わなければならず、その分実行時間が長くなる（実行開始が遅れる）。

【００２０】

次に、マルチスレッド制御装置１の構成を説明する。

マルチスレッド制御装置１は、スレッド管理処理部１１、スレッド回収処理部１２、スレッド管理テーブル１３を含む。

【００２１】

スレッド管理処理部１１は、スレッド生成をＯＳに要求する処理、及びスレッドにプログラムの実行を要求する処理を実行する。スレッド回収処理部１２は、待機スレッドを監視し、無駄な待機スレッドを回収（スレッド終了）する処理を実行する。スレッド管理テーブル１３は、スレッド情報を管理するためのテーブルである。

【００２２】

尚、図１には、ＯＳ２にて生成されたスレッドとして３つのスレッド１４，１５，１６を示す。これらスレッド１４～１６のうち、スレッド１４，１５はプログラムを実行している実行中スレッドであり、スレッド１６はプログラムを終了して待機している待機スレッドとして両処理部１１，１２の動作及びテーブル１３の構成を説明する。

【００２３】

先ず、スレッド管理テーブル１３の構成を図３，４に従って説明する。

図４は、スレッド管理テーブル１３の概略構成図である。

スレッド管理テーブル１３は、情報管理テーブル２１と複数（図４では、図１のスレッド１４～１６に対応して３つ）のスレッド制御表２２～２４を含む。情報管理テーブル２１は、実行中スレッドキュー２１ａ、待機スレッドキュー２１

ｂ、待機スレッドカウンタ２１ｃ、実行中スレッドカウンタ２１ｄ、最大実行中スレッドカウンタ２１ｅのそれぞれのための領域を持つ。

【００２４】

実行中スレッドキュー２１ａは、それにキューイングされた実行中スレッドに対応するスレッド制御表を指すアドレスを記憶する領域を持つスレッド制御表アドレスリストであり、実行中のスレッドの情報を管理する。待機スレッドキュー２１ｂは、それにキューイングされた待機スレッドに対応するスレッド制御表を指すアドレスを記憶する領域を持ち、待機状態のスレッドの情報を管理する。詳述すると、図１においてスレッド１４，１５が実行中スレッド、スレッド１６が待機スレッドである。これに対し、実行中スレッドキュー２１ａには実行中スレッド１４，１５に対応するスレッド制御表２２，２３を指すアドレスが記録され、待機スレッドキュー２１ｂには待機スレッド１６に対応するスレッド制御表２４を指すアドレスが記録されている。

【００２５】

図３は、スレッド制御表２４の構成を示す説明図である。尚、他のスレッド制御表２２，２３の構成はスレッド制御表２４のそれと同じであるため、図面及び説明を省略する。

【００２６】

スレッド制御表２４は、スレッド実行ＥＣＢ(Event Control Block) ２４ａ、スレッド停止ＥＣＢ２４ｂのための領域を持つ。スレッド実行ＥＣＢ２４ａは図１のスレッド管理処理部１１からの実行要求を管理するために設けられ、スレッド停止ＥＣＢ２４ｂは図１のスレッド回収処理部１２からの終了要求を管理するために設けられている。図１の待機スレッド１６は、各ＥＣＢ２４ａ，２４ｂに書き込まれた情報に基づいてプログラムの実行又は停止処理を行う。

【００２７】

詳述すると、図１のスレッド管理処理部１１は、待機スレッドキュー２１ｂにキューイングされたスレッド制御表２４のスレッド実行ＥＣＢ２４ａに実行要求に応じた情報を格納する。待機スレッド１６は、自己に対応したスレッド制御表２４のスレッド実行ＥＣＢ２４ａに実行要求に応じた情報が格納されると、それ

に応答してスレッドに割り当てられたプログラム部分を実行する。

【００２８】

図１のスレッド回収処理部１２は、待機スレッドキュー２１ｂにキューイングされたスレッド制御表２４のスレッド停止ＥＣＢ２４ｂに停止要求に応じた情報を格納する。待機スレッド１６は、スレッド停止ＥＣＢ２４ｂに停止要求に応じた情報が格納されると、それに応答して終了処理を実行する。これにより、待機スレッド１６に割り当てられていたシステム資源が解放される。

【００２９】

待機スレッドカウンタ２１ｃには、待機状態のスレッド数が記録され、実行中スレッドカウンタ２１ｄには、実行中のスレッド数が記録され、最大実行中スレッドカウンタ２１ｅには実行中のスレッド数の最大値が記録される。

【００３０】

図１の実行中スレッド１４，１５は、プログラムを終了すると、実行中スレッドカウンタ２１ｄをカウントダウン（－１）し、待機スレッドカウンタ２１ｃをカウントアップ（＋１）する。図１の待機スレッド１６は、スレッド管理処理部１１から実行要求を受け取ると、待機スレッドカウンタ２１ｃをカウントダウンし、実行中スレッドカウンタ２１ｄをカウントアップし、そのカウンタ２１ｄの値が最大実行中スレッドカウンタ２１ｅの値よりも大きい場合、最大実行中スレッドカウンタ２１ｅの値を実行中スレッドカウンタ２１ｄの値に更新する。このように、両カウンタ２１ｃ，２１ｄは、待機中のスレッド数と実行中のスレッド数を表す。

【００３１】

従って、図１のスレッド管理処理部１１は、待機スレッドカウンタ２１ｃに基づいて、スレッド生成要求、スレッドに対する実行要求を行う。詳述すると、スレッド管理処理部１１は、並列のプログラムの実行要求を受けると、待機スレッドカウンタ２１ｃが「０」の場合にはＯＳ２にスレッド生成を要求し、生成されたスレッドを実行中スレッドキュー２１ａにキューイングし、そのスレッドにプログラムの実行を要求する。一方、カウンタ２１ｃが「１」以上の場合、スレッド管理処理部１１は、待機スレッドキュー２１ｂにキューイングされた待機スレ

ッド１６にプログラムの実行を要求する。

【００３２】

また、図１のスレッド回収処理部１２は、待機スレッドカウンタ２１ｃと最大実行中スレッドカウンタ２１ｅの値に基づいて、余分なスレッドを回収する。詳述すると、スレッド回収処理部１２は、最大実行中スレッドカウンタ２１ｅと待機スレッドカウンタ２１ｃを所定の時間毎にモニタし、それらを比較する。そして、スレッド回収処理部１２は、比較結果に基づいて、最大実行中スレッドカウンタ２１ｅの値より待機スレッドカウンタ２１ｃのそれの方が大きい場合に、最大実行中スレッドカウンタ２１ｅの値まで待機スレッドを回収する。即ち、スレッド回収処理部１２は、両カウンタ２１ｃ，２１ｅの値の差の数のスレッドに対して終了要求を行い、待機スレッドを回収する。

【００３３】

次に、スレッド１４～１６及びスレッド回収処理部１２の動作を図５，図６のフローチャートに従って説明する。

図５は、スレッドが実行する処理を説明する動作フローチャートである。

【００３４】

先ず、ステップ３１において、スレッドは、実行中スレッドカウンタ２１ｄを＋１する。

次に、ステップ３２において、スレッドは実行中スレッドカウンタ２１ｄと最大実行中スレッドカウンタ２１ｅの値を比較し、実行中スレッドカウンタ２１ｄの方が最大実行中スレッドカウンタ２１ｅより大きい場合はステップ３３へ移る。そのステップ３３において、スレッドは最大実行中スレッドカウンタ２１ｅの値を実行中スレッドカウンタ２１ｄのそれに更新する。

【００３５】

ステップ３４において、スレッドはプログラムを実行する。そのプログラムが終了すると、スレッドはステップ３５に移る。

ステップ３５において、スレッドは実行中スレッドカウンタ２１ｄを－１する。次に、ステップ３６において、スレッドは待機スレッドカウンタ２１ｃを＋１する。そして、ステップ３７において、スレッドは待機スレッドキュー２１ｂに

スレッド制御表をキューイングする。

【００３６】

ステップ３８において、スレッドはマルチＷＡＩＴ処理を実行する。この処理において、スレッドは、スレッド実行ＥＣＢ２４ａとスレッド停止ＥＣＢ２４ｂの両方を監視し、何れか一方に情報が書き込まれるまで待つ。そして、スレッドは、スレッド実行ＥＣＢ２４ａまたはスレッド停止ＥＣＢ２４ｂに情報が書き込まれると、ステップ３９に移る。

【００３７】

ステップ３９において、スレッドは、ステップ３８において書き込まれた情報が終了要求か否かを判断し、終了要求ではない、即ち実行要求の場合にはステップ４０に移る。そのステップ４０において、スレッドは、待機スレッドカウンタ２１ｃを－１した後、ステップ３１に移る。これにより、スレッドはプログラムを実行するスレッドとして機能し、待機スレッドが再利用されることになる。

【００３８】

一方、ステップ３９において、入力が終了要求の場合、スレッドはステップ４１に移り、そのステップ４１において終了処理を実行する。これより、スレッドが終了（消滅）する。

【００３９】

図６は、スレッド回収処理部１２の動作フローチャートである。

先ず、ステップ５１において、スレッド回収処理部１２は、所定時間のタイマをかける。この所定時間は、待機スレッドカウンタ２１ｃと最大実行中スレッドカウンタ２１ｅをモニタする間隔であり、このモニタ間隔はシステムに応じて設定した一定値、又は稼働時刻等に対応して変更してもよい。そして、所定時間経過してタイムアップすると、スレッド回収処理部１２はステップ５２に移る。

【００４０】

ステップ５２は比較処理（比較手段）であり、スレッド回収処理部１２は、待機スレッドカウンタ２１ｃと最大実行中スレッドカウンタ２１ｅとを比較し、待機スレッド数が最大実行中スレッド数より大きくなっていないか確認する。そして、スレッド回収処理部１２は、待機スレッド数より最大実行中スレッド数の方

が大きい場合にはステップ５１に戻り、待機スレッド数の方が最大実行中スレッド数より大きい場合にはステップ５３に進む。

【００４１】

そのステップ５３は回収処理（回収手段）であり、スレッド回収処理部１２は、待機スレッドに終了要求を行い（詳しくは待機スレッドに対応するスレッド制御表のスレッドＥＣＢ２４ｂに終了要求に応じた情報を書き込む）、スレッドを回収する。

【００４２】

上記マルチスレッド制御装置１は、計算機上で実行するコンピュータプログラムにより実現することもできる。

即ち、図２に示すように、マルチスレッドにてプログラムを並列実行する計算機６０は、処理装置６１、入出力装置６２、主記憶装置６３、補助記憶装置６４などから構成され、コンピュータプログラムを実行するものである。コンピュータプログラムは、フロッピーディスクやＣＤ－ＲＯＭ等の可搬型記録媒体６５やネットワーク接続された他の計算機の主記憶装置や補助記憶装置等に格納されて提供される。

【００４３】

提供されたコンピュータプログラムは、可搬型媒体６５から一旦補助記憶装置６４にコピーまたはインストール後に主記憶装置６３にロードされ、または可搬型媒体６５から直接計算機６０の主記憶装置６３にロードされ実行する。尚、主記憶装置６３には、図１のスレッド管理テーブル１３が作成され、それによりスレッド情報が管理される。

【００４４】

また、コンピュータプログラムがネットワーク接続された他の装置に格納されて提供された場合も、他の装置からネットワークを経由して受信後に補助記憶装置６４にコピーまたはインストール後に主記憶装置６３にロードされ、または直接主記憶装置６３にロードされ実行するものである。

【００４５】

尚、図２は、コンピュータプログラムを実行する計算機を機能的に示したもの

であり、各装置６１～６４が複数設けられた計算機を用いる、又は複数の計算機が疎又は密結合された計算機システムを用いても良い。

【００４６】

以上記述したように、本実施の形態によれば、以下の効果を奏する。

（１）マルチスレッド制御装置１のスレッド回収処理部１２は、実行中のスレッドの数を所定期間監視し、その期間内の最大値（最大実行中スレッド数）と待機スレッド数とを比較する。そして、スレッド回収処理部１２は、最大実行中スレッド数よりも待機スレッド数が大きい場合、最大実行中スレッド数まで待機スレッドを回収するようにした。その結果、無駄にシステム資源を使用している待機スレッドを回収し、実行中のスレッドに与える影響を少なくすることができる。

【００４７】

（２）スレッド回収処理部１２は、最大実行中スレッドカウンタ２１ｅの値、即ち所定期間に於ける実行中スレッド数の最大値まで待機スレッドを回収する。この最大実行中スレッド数は、実行しているプログラムの状態、即ち、システムの稼働状況に応じて変化する。その結果、最大実行中スレッド数まで待機スレッドを回収することで、システムの稼働状況に応じた数の待機スレッドを待機させることができる。

【００４８】

尚、前記実施形態は、以下の態様に変更してもよい。

○上記実施形態では、所定時間毎に待機スレッドカウンタ２１ｃと最大実行中スレッドカウンタ２１ｅを比較した結果に基づいて待機スレッドを終了させるようにしたが、待機スレッドカウンタ２１ｃの値をその時における待機スレッドの必要数とし、それと実行中スレッドカウンタ２１ｄを比較した結果に基づいて待機スレッドを回収するようにしても良い。また、必要数として所定期間に於ける実行中スレッド数の平均値を求め、その平均スレッド数と待機スレッド数とを比較し、その比較結果に基づいて平均値又は現在実行中のスレッド数まで待機スレッドを回収するようにしてもよい。また、実行中スレッド数にシステムの状況に応じた値又は一定値の係数を演算した数を必要数とし、それまで待機スレッドを

回収するようにしてもよい。

【００４９】

○上記実施形態では、最大実行中スレッドカウンタ２１ｅの更新を各スレッド１４～１６が行うようにしたが、その更新処理をスレッド管理処理部１１又はスレッド回収処理部１２が行うようにしてもよい。

【００５０】

【発明の効果】

以上詳述したように、本発明によれば、実行中のスレッドの数と待機状態にあるスレッドの数に基づいて不要な数の待機スレッドを回収するようにした。これにより、待機スレッドにてプログラム実行速度の高速化を図り、無駄にメモリ資源を使用している待機スレッドを終了させ、システム資源の有効活用を図ることができる。

【００５１】

また、所定時間毎に、必要数まで前記待機スレッドを回収するようにしたため、稼働状況に応じた必要数にまで待機スレッドの数を低減してシステム資源の有効活用を図ることができる。

【図面の簡単な説明】

【図１】　一実施形態のマルチスレッド制御装置の概略構成図である。

【図２】　システムのハードウェア構成を示す概略図である。

【図３】　スレッド制御表の説明図である。

【図４】　スレッド管理テーブルの説明図である。

【図５】　スレッドの処理を示すフローチャートである。

【図６】　スレッド回収処理部の処理を示すフローチャートである。

【符号の説明】

１１　スレッド管理処理部

１２　スレッド回収処理部

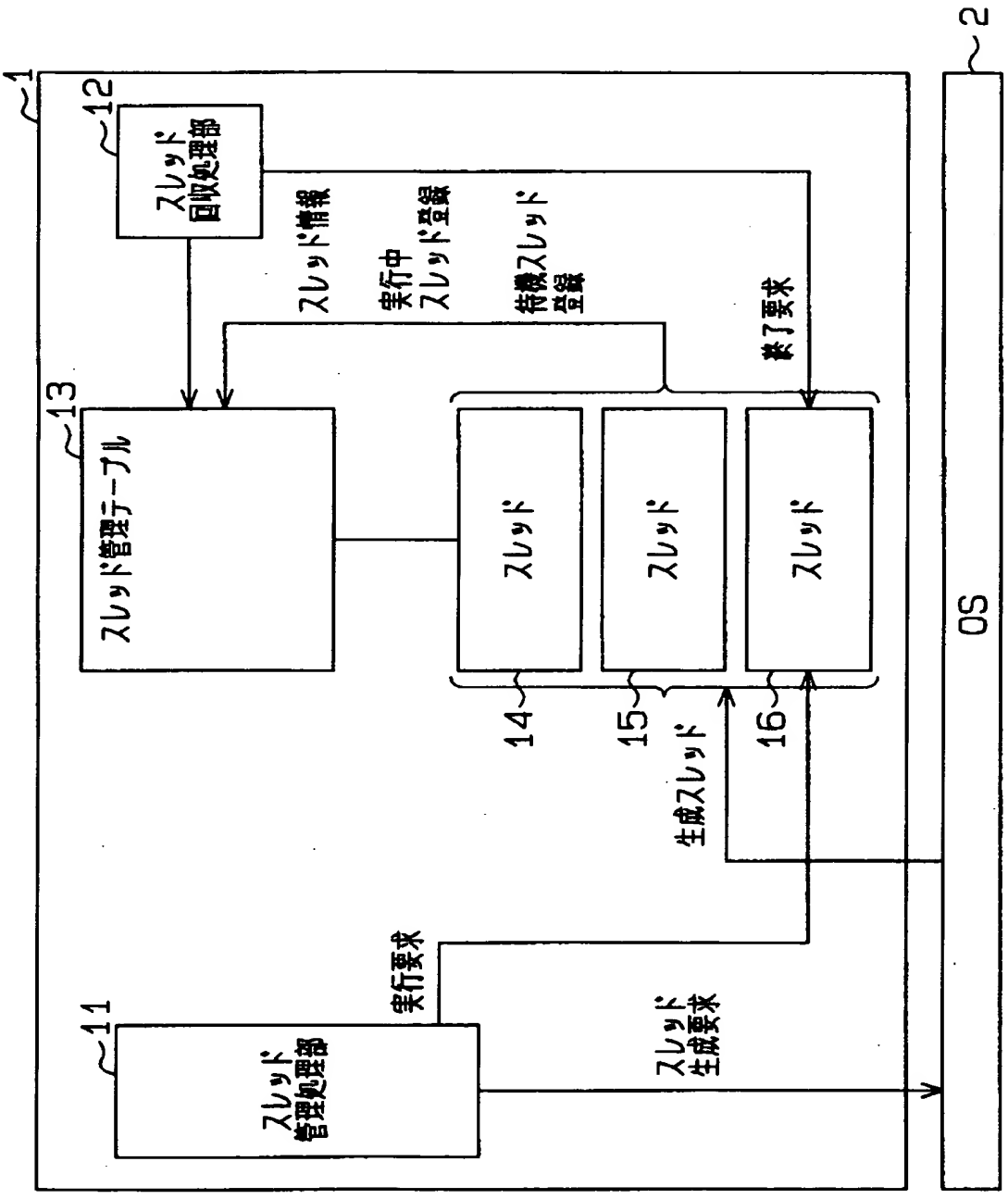
１３　スレッド情報管理部としてのスレッド管理テーブル

１４～１６　スレッド

【書類名】　　図面

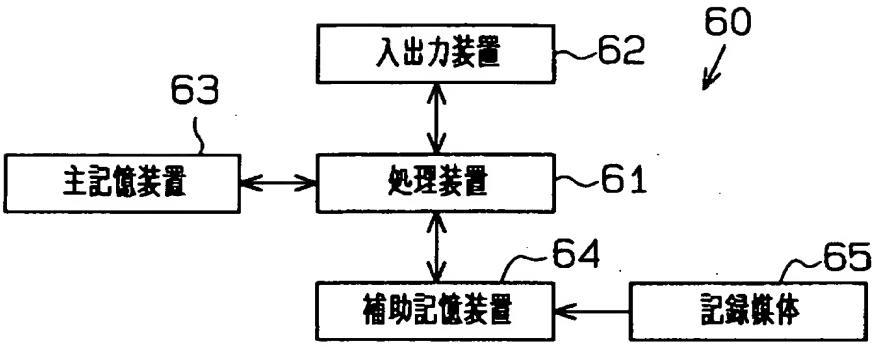
【図１】

マルチスレッド制御装置の概略構成図



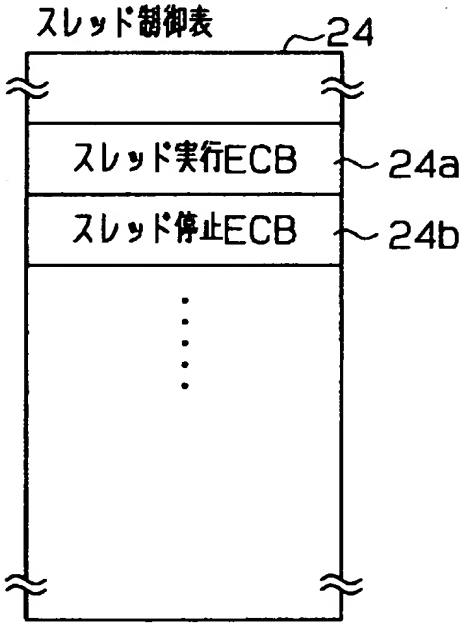
【図２】

システムのハードウェア構成を示す概略図



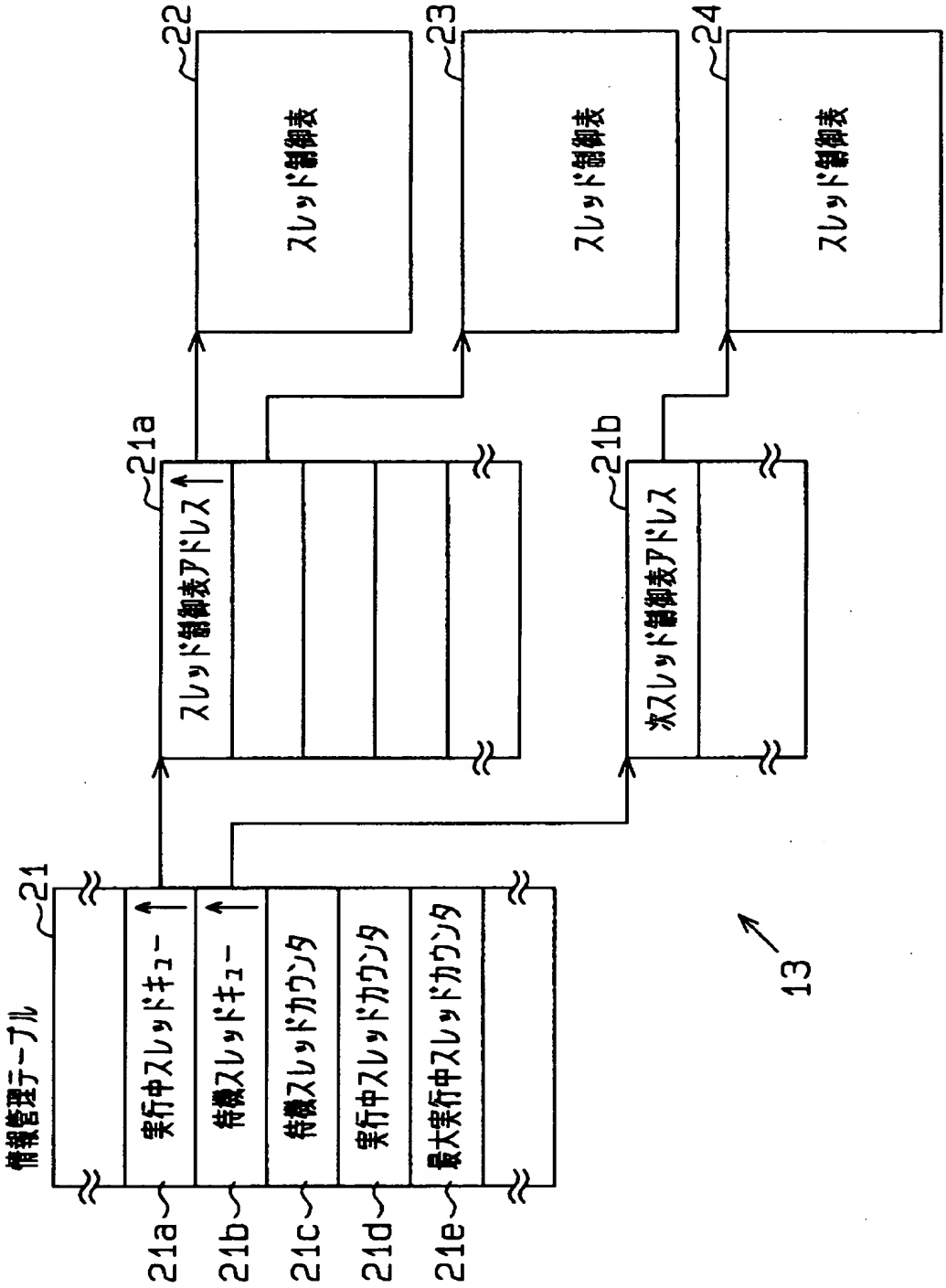
【図３】

スレッド制御表の説明図



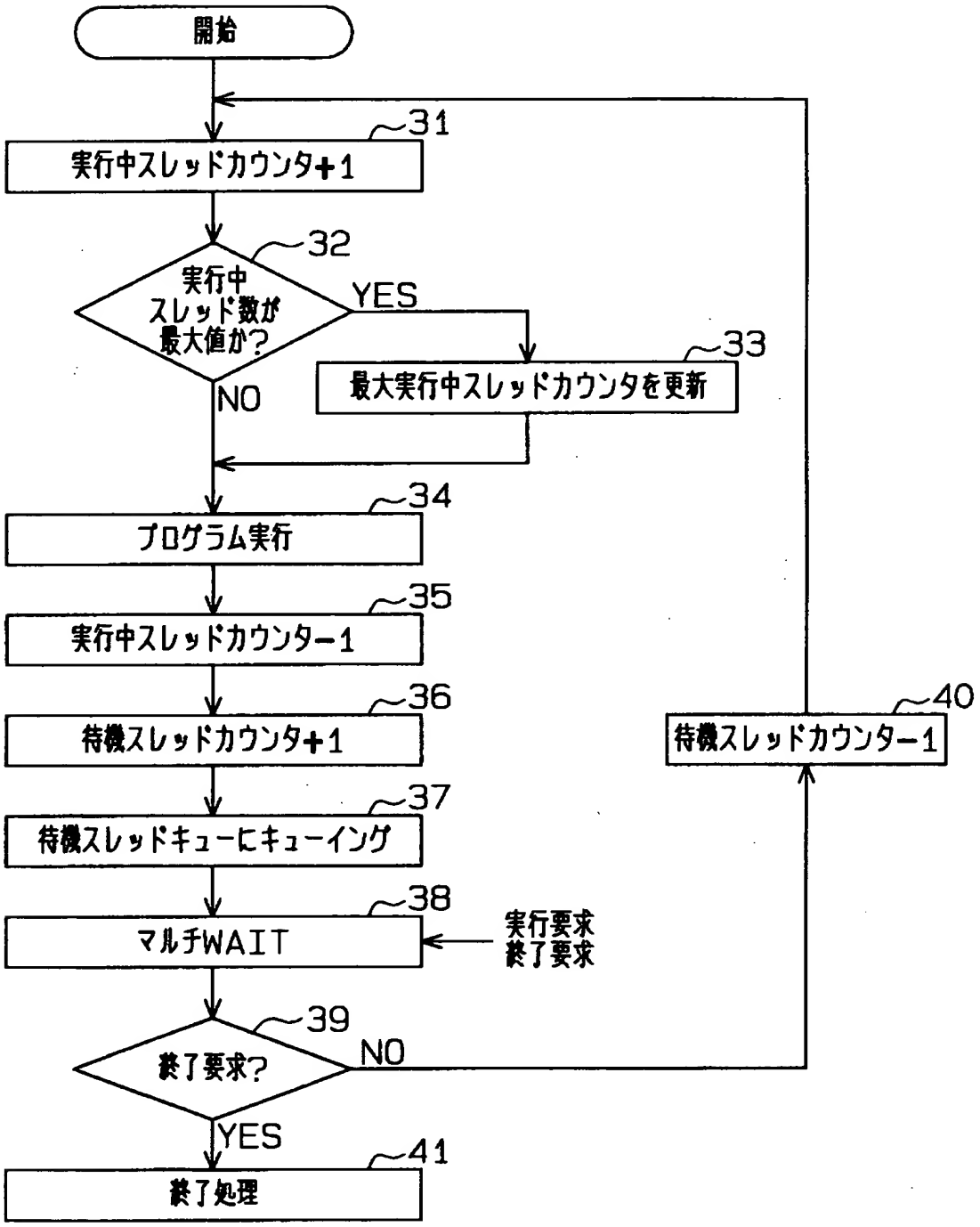
【図４】

スレッド管理テーブルの説明図



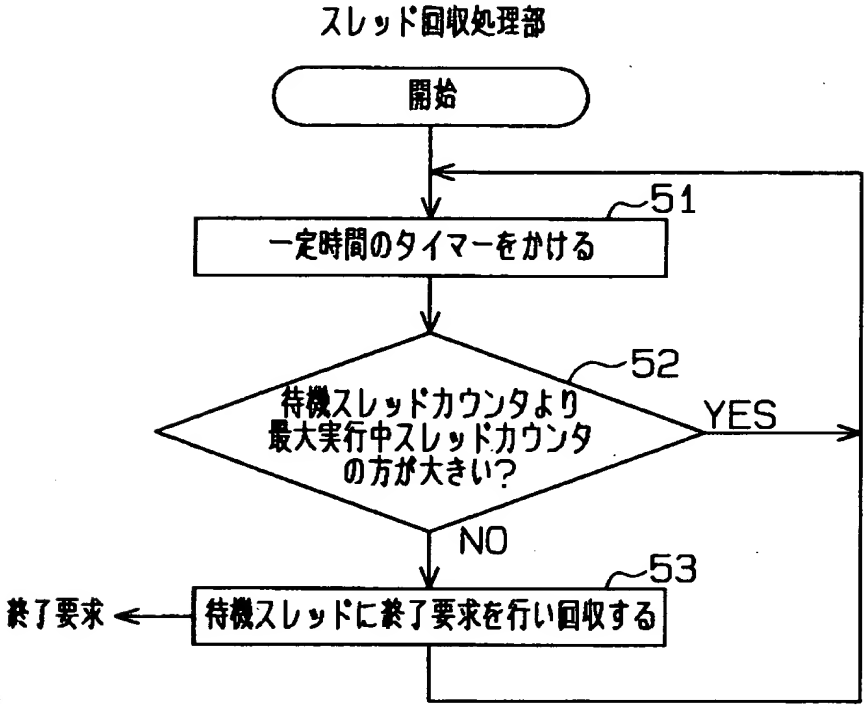
【図５】

スレッドの処理を示すフローチャート



【図６】

スレッド回収処理部の処理を示すフローチャート



【書類名】　　要約書

【要約】

【課題】プログラム実行速度の高速化とシステム資源の有効活用を図ることのできるマルチスレッド制御装置を提供すること。

【解決手段】マルチスレッド制御装置１のスレッド回収処理部１２は、実行中のスレッドの数を所定期間監視し、その期間内の最大値（最大実行中スレッド数）と待機スレッド数とを比較する。そして、スレッド回収処理部１２は、最大実行中スレッド数よりも待機スレッド数が大きい場合、最大実行中スレッド数まで待機スレッドを回収する。

【選択図】　　図１

出　願　人　履　歴　情　報

識別番号　　〔０００００５２２３〕

１．変更年月日　　１９９６年　３月２６日
［変更理由］　住所変更
住　所　神奈川県川崎市中原区上小田中４丁目１番１号
氏　名　富士通株式会社